ARMY RESEARCH LABORATORY

The Definition and Ray-Tracing of B-Spline Objects in a Combinatorial Solid Geometric Modeling System

by Paul R. Stay

ARL-RP-428 **April 2013**

A reprint from the *Fourth USENIX Computer Graphics Workshop*, Cambridge, MA, 9 October 1987.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5068

ARL-RP-428 **April 2013**

The Definition and Ray-Tracing of B-Spline Objects in a Combinatorial Solid Geometric Modeling System

Paul R. Stay
Survivability/Lethality Analysis Directorate, ARL

A reprint from the *Fourth USENIX Computer Graphics Workshop*, Cambridge, MA, 9 October 1987.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*: April 2013	**2. REPORT TYPE**: Reprint
3. DATES COVERED (From - To): October 1987	
4. TITLE AND SUBTITLE: The Definition and Ray-Tracing of B-Spline Objects in a Combinatorial Solid Geometric Modeling System	**5a. CONTRACT NUMBER**:
	5b. GRANT NUMBER:
	5c. PROGRAM ELEMENT NUMBER:
6. AUTHOR(S): Paul R. Stay	**5d. PROJECT NUMBER**:
	5e. TASK NUMBER:
	5f. WORK UNIT NUMBER:
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES): U.S. Army Research Laboratory ATTN: RDRL-SLB-S Aberdeen Proving Ground, MD 21005-5068	**8. PERFORMING ORGANIZATION REPORT NUMBER**: ARL-RP-428
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES):	**10. SPONSOR/MONITOR'S ACRONYM(S)**:
	11. SPONSOR/MONITOR'S REPORT NUMBER(S):

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

A reprint from the *Fourth USENIX Computer Graphics Workshop*, Cambridge, MA, 9 October 1987.

14. ABSTRACT

Traditionally there has been a distinction between Combinatorial Solid Geometry (CSG) modeling systems and Sculptured Surface Design modeling systems. CSG modeling systems largely model parts which are unsculptured and consist of combinations of common shapes like spheres, prisms, ellipsoids, and the like. These shapes are represented as planar half spaces, and algebraic quadratic surfaces. The boolean combination of these surfaces is usually performed by ray-tracing. Sculptured Surface Design concerns itself with modeling the surface of an object, i.e., the boundaries of an object like an aircraft, a ship, or an automobile. The boundaries are represented by using parametric tensor-product surfaces consisting of Bezier curves and Nonuniform Rational B-spline Surfaces (NURBS). There are many times however, when both modeling approaches are needed. In particular it is often desirable to introduce free-form surfaces into the CSG system. Recent advances in ray-tracing free-form surfaces have allowed the integration of free-form objects in CSG systems. This presentation will discuss the development and integration of NURBS into the Ballistics Research Laboratory CSG modeling system.

15. SUBJECT TERMS

NURBS, BSpline, raytracing, CSG, BRL-CAD

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
Unclassified	Unclassified	Unclassified	UU	12	Clifford W. Yapp
					19b. TELEPHONE NUMBER (*Include area code*): 410-278-1382

The Definition and Ray-tracing of B-spline Objects in a Combinatorial Solid Geometric Modeling System

Paul Randal Stay

Advance Computer Systems Team
US Army Ballistic Research Laboratory
Aberdeen Proving Ground
Maryland 21005-5066 USA

ABSTRACT

Traditionally there has been a distinction between Combinatorial Solid Geometry (CSG) modeling systems and Sculptured Surface Design modeling systems. CSG modeling systems largely model parts which are unsculptured and consist of combinations of common shapes like spheres, prisms, ellipsoids, and the like. These shapes are represented as planar half spaces, and algebraic quadratic surfaces. The boolean combination of these surfaces is usually performed by ray-tracing. Sculptured Surface Design concerns itself with modeling the surface of an object, i.e., the boundaries of an object like an aircraft, a ship, or an automobile. The boundaries are represented by using parametric tensor-product surfaces consisting of Bezier curves and Nonuniform Rational B-spline Surfaces (NURBS). There are many times however, when both modeling approaches are needed. In particular it is often desirable to introduce free-form surfaces into the CSG system. Recent advances in ray-tracing free-form surfaces have allowed the integration of free-form objects in CSG systems. This presentation will discuss the development and integration of NonUniform Rational B-splines into the BRL CSG modeling system.

1. Introduction.

Computer Aided Geometric Design, since its beginning in the mid-1960's has taken two different approaches to the modeling of mechanical parts and objects: sculptured surface design and combinatorial solid geometry (also known as volumetric solid modeling). Each approach was developed to represent different types of objects and requires a different style of object definition. Sculptured surface design concerns itself with modeling the surface boundaries (i.e., an aircraft or ship hull). CSG systems model parts that are unsculptured and consist of combinations of common shapes like spheres, prisms, cones, and the like.

Free-form surfaces are hard to represent as boolean combinations of these volumetric solids, therefore a faceted polyhedron was introduced to allow for a rough approximation of the surface geometry. Faceted polyhedra are useful in many applications and analyses that require a minimum amount of surface geometry. However, geometric information such as gaussian curvature requires a more accurate description of the surface geometry than is currently available using a faceted approach. There are many problems associated with the extension of CSG modeling systems to include sculptured surface primitives. The addition of these free-form surfaces, independent of their representation, requires the introduction of a new surface object type to the CSG system. These free-form object types can be defined by using either an implicit mathematical representation (e.g. superquadrics) or a boundary representation. In the case of "superquadrics" the modeling system does not inherit a well developed sculptured surface form. The same style of inside and

outside determination cannot readily be made with the boundary model as with the procedural solid primitive representations, hence problems arise in implementing the boolean combinations of the solids made with sculptured surfaces.

2. Techniques for Rendering Boolean combinations of surfaces.

One approach, applied by the University of Utah Alpha_1 project is to treat all objects as sculptured surfaces represented as tensor product Nonuniform Rational B-spline Surfaces (NURBS).[1] Solid volumes are represented by a collection of B-spline surfaces called a shell. A key ingredient of the Alpha_1 system is the Oslo algorithm[2] which provides a computational technique for subdividing B-spline surfaces. Using the algorithm, Spencer Thomas [3] has defined and implemented a classification scheme that allows boolean operations to be performed on sculptured surfaces. An interesting sidelight of this intersection algorithm is that B-spline surfaces do not need to describe a closed volume, allowing for the ability to have partially bounded sets.

Since NURBS are the fundamental representation, each of the CSG solids can be derived and represented with other defined volumetric primitives such as rounded edge boxes made of collections of B-spline surfaces.[4,5] There are two drawbacks to this approach however. First representing a sphere as a NURB may not be as efficient as a CSG representation. Secondly the representation of the intersection of two B-spline surfaces is not a B-spline surface but a collection of polygons.

Another approach, used by the Reyes image rendering system,[6] involves an extended Z-buffer algorithm which stores multiple z values for each solid to allow boolean operations between objects.

For the past 20 years, the Ballistic Research Laboratory has been using boolean combinations of simple volumetric shapes to design and analyze US Army vehicles.[7] Ray/solid intersection algorithms generate line segments that are used to classify the solids for boolean combinations. New advances in ray-tracing [8,9] show it is possible to calculate ray intersections with tensor product NURBS. This gives the capability to represent free-form surfaces in a CSG modeling system with additional surface geometric information and allows boolean combinations between solids in the system. Since ray-tracing is required for many of the applications within the BRL CSG modeling system, the ray/B-spline intersection algorithm has been integrated into that system.

3. B-spline Solid Definitions.

Tensor product Nonuniform Rational B-spline Surface properties have been discussed in a number of papers [10,11,12] and there have been a number of modelers written to edit splines. No attempt will be made here to discuss the different approaches in modeling systems that are used to create and manipulate NURBS.

A B-Spline solid can be defined as a collection of tensor product Nonuniform Rational B-spline Surfaces. These surfaces are used to define the boundary of the volume which is to be represented. However, there are constraints that must be met if NURBS or any other boundary representation is to be integrated into a CSG system.

Since all ray/solid intersections are required to perform boolean operations between surfaces, each surface or collection of surfaces must completely enclose space. Surfaces which are joined need to be specified such that the common boundary curves exactly match and that no gaps exist. This is required to ensure that the primitive represents a solid.

Surface normals of the NURB solid are required to point outward. This guarantees that the boolean operations and applications (such as rendering) result in solids which are consistent within the CSG system.

4. Ray-trace Algorithm

There have been many different approaches proposed for the ray/B-spline intersection algorithm. The most notable of these use either Newton's iteration method for determining the intersection point,[8,9] or tessellate the surface into a polygonal mesh. [13] Techniques which use the Newton iteration method tend to be computationally intensive, but do not lose the topology of the B-

spline surface. While less computationally intensive than the Newton iteration method, techniques that subdivide the surface into polygons tend to lose the topology of the B-spline surfaces.

The ray/B-spline solid intersection routines used in the BRL CAD ray-trace library[14] are based on techniques that were outlined in the original Oslo algorithm paper. Since the B-spline surface lies within the convex hull of the control mesh, a bounding box of the surface can be described by taking the minimum and the maximum of the de Boor net. The subdivision is performed by adding order multiple knots at the parametric midpoint in one of the given directions. The result of the subdivision is two distinct B-spline surfaces that represent the original surface. The extent of subdivision is determined by the following conditions, with more subdivision necessary if the condition(s), checked in the order listed, exist.

1. The ray intersects the bounding box of the convex hull boundary of the surface.
2. Interior knots of the B-spline exist.
3. The surface is not flat according to some flatness criteria.

Since ray-tracing is performed in object space, traditional scan line techniques for determining a flatness parameter for the surface are invalid. Flatness testing of the surface uses a modified form of cone and beam tracing. Each ray that is generated by the application program is given an initial beam radius r and a slope of beam divergence per millimeter s. One of the results of the ray/bounding box intersection is a parametric distance t from the ray origin to the bounding box of the surface. A variance parameter v is calculated by $v=r+st$ which is used to test the subdivided surface. Points from each row and column of the control mesh are then used to test for flatness. l_i is a line segment which is defined by two distinct points of the row/column, and d_j is the distance of each individual point to the line segment. If the condition $d \leq v$ is true, then the row is determined to be flat. If all rows and columns of the control mesh are flat then a further test is performed on the bezier points of the subdivided patch. A plane is formed from three of the surface corner points. If the distance from the fourth corner point to the plane is $\leq v$ then the surface is determined to be flat.

When a surface is determined to be flat, the four corner points of the control mesh are used to create two polygons which are then intersected with the ray.

If a ray intersects the bounding box of a B-spline surface, then the surfaces are recursively subdivided and tested against the ray until surface flatness criteria are reached or the ray misses the surface. The algorithm is as follows:

```
for each (surface in the B-spline Solid) add surface onto active node list
while (surfaces exist in the active list)
        Get first surface on the active list:
        if (the ray intersects the bounding box of the surface)
                If (the surface is flat)
                        intersect ray with the polygons
                        and sort hit point into the hit list.
                else (the surface is not flat)
                        subdivide the surface and insert the
                        two returned surfaces on the active list.
        else remove from active list.
continue until no surfaces exist on the active list
```

The sorted hit points are used to create line segments that describe the ray/solid intersection. All line segments are collected and the boolean operations are performed on all solid segments.

The B-spline surface subdivision tends to be computationally expensive to perform on conventional computers. However, the algorithm can be optimized by generating and storing the bounding boxes and the subdivided surfaces in a binary tree. The ray can then be tested recursively against the stored binary tree. Subdivision of the B-spline surfaces is performed at the time of the ray intersection testing, thus only those portions of the tree that were intersected by a ray are generated.

Many of the ray-tracing applications at the BRL need to calculate principle curvature in each direction on a surface. A method of calculating the derivatives of a B-spline surface using the control points[2] can be used for non rational B-splines.

5. Rational B-Spline surfaces.

Rational B-spline surfaces are used to exactly represent conic sections such as ellipsoids and hyperbolas and are important to aircraft designers. The rational B-spline is defined as:

$$S(u,v)=\left(\frac{x(u,v)}{\omega(u,v)}, \frac{y(u,v)}{\omega(u,v)}, \frac{z(u,v)}{\omega(u,v)}\right)$$

The Oslo algorithm can be applied to both the numerator as well as the denominator. The ω values are weights assigned to each of the points in the control mesh and can be represented in homogeneous space.[15] Rational surfaces which pass the flatness test divide the ω values of the corners from the control mesh to form the 3 space polygon points, which are passed to the ray/polygon intersection routine.

Rational surface must be treated separately for the calculation of curvature since the quotient rule must be applied. The formula for calculating the derivative of a rational B-spline in the u parametric direction is:

$$\frac{\partial S}{\partial u}=\frac{\omega(u,v)\left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}\right) - (x(u,v),\ y(u,v),\ z(u,v))\frac{\partial \omega}{\partial u}}{\omega(u,v)^2}$$

There is hope that the computation can be made a bit more reasonable. Essentially one can still use the de Boor algorithm with the homogenous points in the control mesh. Substitution can then be used to calculate the derivative values. Thus, the $\frac{\partial S}{\partial u}$ can be expressed as follows:

$$\frac{\partial S}{\partial u} = \frac{1}{w(u,v)}\left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}\right) - \frac{\frac{\partial \omega}{\partial u}}{\omega(u,v)^2}\,(x(u,v),\ y(u,v),\ z(u,v))$$

Similar expressions can be calculated for the rest of the derviatives for calculating the principle curvature.

6. Future Work.

Research in new computer hardware and software techniques should improve the speed of the ray/B-spline intersection.calculations.

In the hardware department, there are sections of the subdivision code that may take advantage of vectorization and parallelization such as on the Alliant and Cray computer systems. Specialized VLSI hardware is being developed by the University of Utah Alpha_1 project that will execute the Oslo algorithm and allow fast subdivision of Nonuniform Rational B-spline Surfaces. The

use of this specialized hardware will not only facilitate faster subdivision of the B-spline surfaces but will allow for some generality in the possible ray/bounding box intersection routines.

There are a number of software optimizations that will be investigated which may improve the algorithm. One area is that of the amount of memory which is necessary to store the binary tree and its subdivided surfaces. Currently the subdivision code returns two surfaces by performing the subdivision in the original surface. Refining the surface instead of splitting it will eliminate the excess data now common between the two surfaces returned from the subdivision algorithm. The routine to check for flatness should be able to return a direction for the subdivision since it can find the area of greatest variance in the control mesh. This will allow the surface to be subdivided in the direction of the larger parametric surface curvature.

7. Acknowledgments.

I would like to thank Peter Stiller who helped with the formulation of the derivatives for both rational and non rational B-splines surfaces. Thanks also go to the people of the Alpha_1 project at the University of Utah who have generated a rich set of B-spline based editing programs and tools. Thanks also to Paul Deitz and Mike Muuss for their influence and ideas.

1. E. Cohen, "Mathematical Tools for a Modelers Workbench," *IEEE Computer Graphics and Applications* (October 1983).

2. R. F. Riesenfeld, E. Cohen, T. Lyche and C. deBoor, "A Practical Guide to Splines," *Computer Graphics and Image Processing*, New York **14**(2), pp. 87-111, Springer-Verlag (1978).

3. S. W. Thomas, *Modelling Volumes Bounded by B-spline Surfaces*, PhD dissertation, University of Utah (June 1984).

4. P. R. Stay, *Rounded Edge Primitives and their Use in Computer Aided Geometric Design*, MS dissertation, University of Utah (August 1984).

5. Pat Hanrahan, "A Survey of Ray-Surface Intersection Algorithms," *SigGraph 87 Course Notes Introduction to Ray Tracing*, Anaheim, California **13** (July 27, 1987).

6. Cook, Carpenter, Catmull, "The Reyes Image rendering Architecture," *Computer Graphics (Proceedings of Siggraph '87)* **21**(4) (July 1987).

7. P. H. Deitz, *Solid Modeling at the US Army Ballistic Research Laboratory*, Proceedings of the 3rd NCGA Conference (13-16 June 1982).

8. M. A. J. Sweeny, R. H. Bartels, "Ray Tracing Free-Form B-spline Surfaces," *IEEE Computer Graphics* (February 1986).

9. John W. Peterson, "Ray Tracing General B-Splines," *Proceedings of the ACM Rocky Mountain Regional Conference*, p. 87 (April, 1986).

10. E. S. Cobb, *Design of Sculptured Surfaces using the B-spline Representation*, PhD dissertation, University of Utah (June 1984).

11. Russ Fish, "Alpha_1:modeling with nonuniform rational b-splines," *Iris Universe* **1**(1) (Spring 1987).

12. Bohm, Farin, Kahmann, "A survey of curve and surface methods in CAGD," *Computer Aided Geometric Desgin* **1**(1) (1984).

13. J. M. Snyder, A H. Barr, "Ray Tracing Complex Models Containing Surface Tessellations," *Computer Graphics (Proceedings of Siggraph '87)* **21**(4) (July 1987).

14. Michael John Muuss, "RT and REMRT shared Memory Parallel and Network Distributed Ray-Tracing Programs," *Proceedings of the Usenix Computer Graphics Workshop* (October 8-9, 1987).

15. W. Tiller, "Rational B-splines for curves and surface representation.," *IEEE Computer Graphics and Applications* **3**(6) (Sept. 1983).

NO. OF COPIES	ORGANIZATION
1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FORT BELVOIR VA 22060-6218
1 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL 2800 POWDER MILL RD ADELPHI MD 20783-1197
1 (PDF)	GOVT PRINTG OFC A MALHOTRA 732 N CAPITOL ST NW WASHINGTON DC 20401
1 (PDF)	USARL RDRL SLE R FLORES WSMR NM 88002-5513
	ABERDEEN PROVING GROUND
1 (HC)	DIR US ARMY EVALUATION CTR HQ TEAE SV P A THOMPSON 2202 ABERDEEN BLVD 2ND FL APG MD 21005-5001
3 (2 HC 1 PDF)	DIR USARL RDRL SL J BEILFUSS P TANENBAUM RDRL SLB A M PERRY (PDF only)
7 (6 HC 1 PDF)	RDRL SLB G KUCINSKI RDRL SLB S S SNEAD (5 CPS) C YAPP (1 PDF)
4 (HC)	QUANTUM RSRCH INTRNTL C HORTON STE 203 2014 TOLLGATE RD BEL AIR MD 21015